**实验报告**

姓名： 专业： 计算机科学与技术 学号：

课程名称： 信息与电子工程导论 任课老师：

实验名称： 基于MATLAB的信号频谱分析 实验日期： 2023.3.5

**1 实验目的和要求**

**1.1 实验目的**

（1）熟悉 matlab 软件的功能使用。

（2）使用 matlab 软件生成含有多个频率分量的信号，并进行频谱分析。

（3）在分析频谱基础之上，对频谱进行滤波，增添部分频率成分等操作。

（4）结合虚拟生成信号，加深的对频谱、傅里叶变换等知识的理解。

**1.2 实验要求**

参考文档《基于MATLAB的信号频谱分析》，学习matlab软件的使用；并自主构建一个至少含有3个频率分量的信号（可以是周期信号，也可以是非周期信号，如含有噪声信号影响）。

进而利用 MATLAB 分析其频谱，并对频谱作适当的处理（滤波操作，或增加一些其他频率成分），观察信号的变化；在此过程中注意采样频率、采样点数对实验结果的影响。

在实验过程中，主要通过matlab可视化绘图功能，展现不同阶段不同信号的变化样式，从而加深对信号时域，频谱，傅里叶变换及其逆变换的理解。

**2 实验原理**

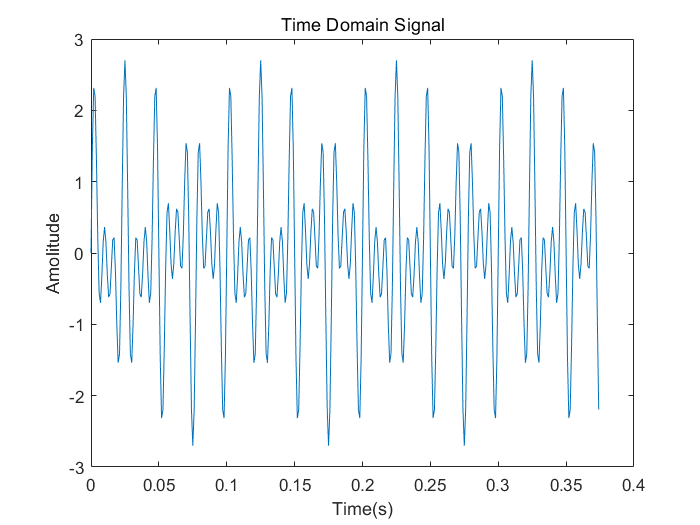
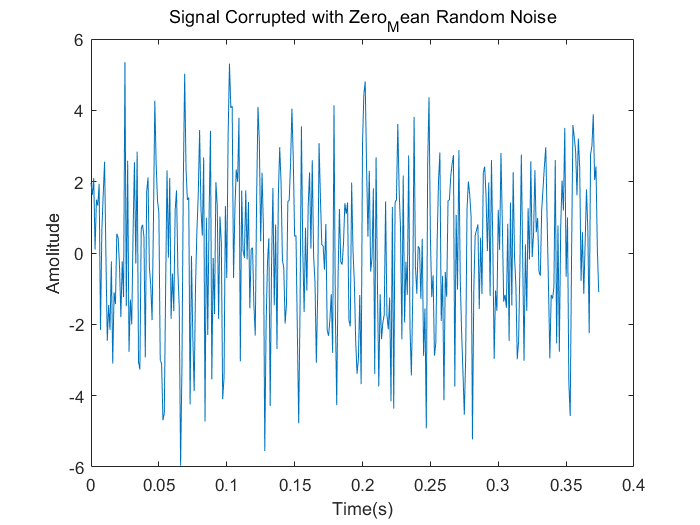
1. matlab软件可以根据数学表达式，借用plot（）函数较为形象的绘制相应的函数曲线，从而满足我们对于信号可视化的需求，便于我们对信号的直观观察。
2. 对于一般的数字信号，我们往往可以使用傅里叶级数展开的方式，将其转换为一系列三角（正弦）函数之和，因此我们对于信号的研究，可以抽象为对正弦函数信号的研究，这为我们自主创造信号产生了便利。
3. 通过绘制“幅度”与“时间”的函数曲线，我们可以得到可视化的信号时域表现图，通过fft（）快速傅里叶变换函数，我们可以获得信号的频域频谱图；通过频谱图，我们可以直观的看到该符合信号的主要频率构成，并可与我们自主设定值进行有效性比对。
4. 在频谱图上，我们可以通过截断的方法，来选择“带宽”，来过滤部分频率，模仿滤波去噪声的特点；在完成对频谱的操作之后，我们可以借助ifft（）快速傅里叶逆变换函数，将频谱图重新转换为时域图，分析截断频率操作对信号数据的影响。

**3 实验内容**

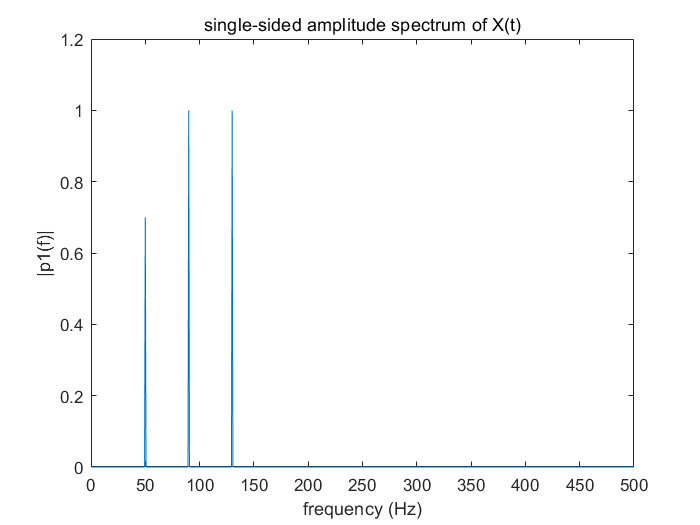
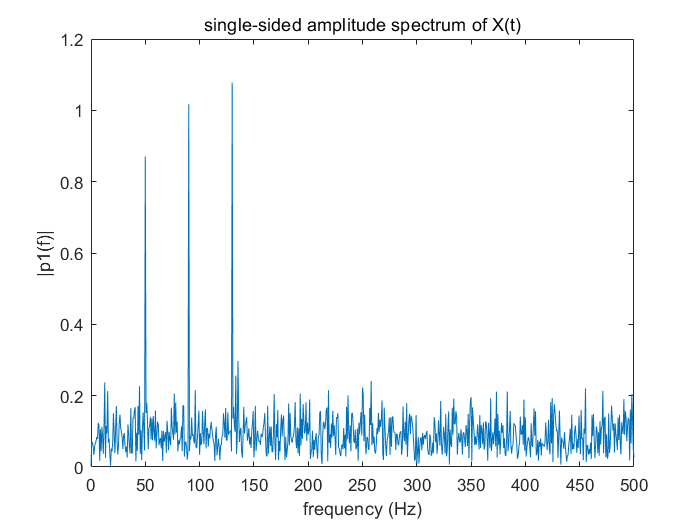
1. 设置信号采样频率与采样长度（采样点数），建立时间函数。
2. 自定义混合频率信号，本实验选择50Hz，90Hz，130Hz的混合信号。
3. 另外创建含有噪声的混合频率信号。
4. 选取采样长度，设置坐标轴与标题信息，分别输出自定义信号与含有噪声信号的时域图，观察幅值与时间的分布关系及噪声对其分布的影响。
5. 使用快速傅里叶变换函数，计算谱密度与单侧频谱幅值，输出自定义信号时域图对映的频谱图。
6. 通过简单的滤波操作，截去小于100Hz的频率信号，并使用快速傅里叶逆变换函数，将相应的频谱图转化为时域图，从而观察信号时域图的变化情况。

**4 实验结果和分析**

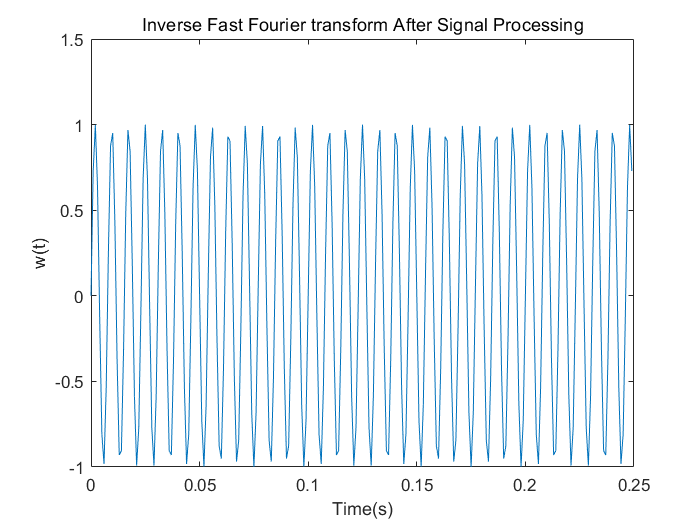
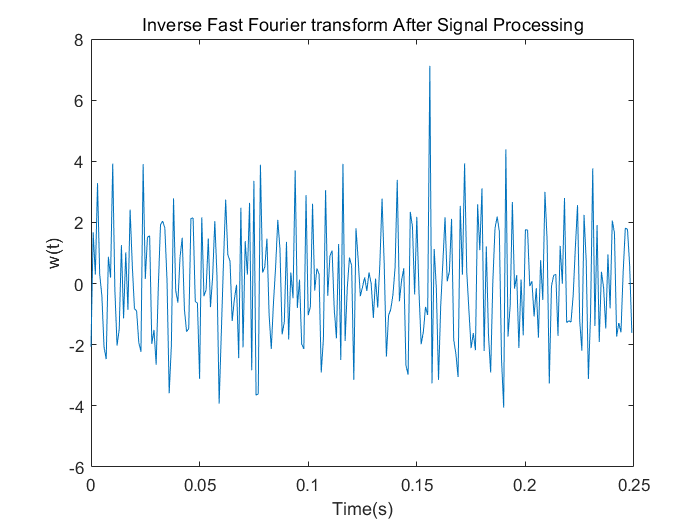
自定义混合频率信号的时域图如下左图所示，可见对于没有噪声影响的信号，其时域图的分布具有明显的周期性。在添加噪声信号后，时域图的绘制结果如下右图所示，与下左图进行对比可以看出，含有噪声的时域图丧失了明显的周期性，因而显得杂乱而没有规律。



经过傅里叶变换后，频谱图的输出结果如下。（左侧为自定义混合信号频谱图，右侧为含有噪声信号的频谱图），经过对比可以明显看出，在没有噪声的影响时，频率的分布特征十分明显；而在加入噪声之后，除了特征频率信号以外，还有大量宽带宽的噪声频率干扰。



进而，我们通过一定的截取方法，滤去小于100Hz的信号，根据自定义混合信号的分析，经过100Hz的滤波效果，不含噪声频率的信号，只剩下130Hz的频率信号，因而在经过快速傅里叶逆变换操作后，信号时域图有极强的周期性与对称性；而对于含有噪声信号而言，时域图虽然减少了部分的复杂度，但是依旧十分复杂，对比结果由下俩图可见（左图为不含噪声频率ifft（）函数处理时域图）。



**5 实验结论**

通过实验，我们可以看到，借助快速傅里叶变换和快速傅里叶逆变换，我们可以很简便且准确地将信号的时域图和频域图进行转化分析；且在信号的DFT变换中，可以清楚的看到DFT变换结果和截取长度之间的关系。

同样，借助于matlab等工具，我们对信号数据的处理变得更加直观与方便；对于一些含有较多噪声的信号，我们也可以通过截取频谱，指定带宽等定量的方法，实现滤波去噪的效果，这对我们后续分析信号有巨大的精确性帮助；另一方面，借助于软件的分析，更是一种模拟的极优途径，使得我们在构建一些新理论，新方法时，有一个虚拟的平台支持我们处理相关的数据，从而分析新建信号理论算法的可行性与可信度，因而，基于matlab的信号频谱分析将成为我们学习“信息与电子工程导论”的重要一环。

**6 源代码与分析**

PS：%后表示代码注释内容

%可选不同采样频率

fs=1000;

%fs=2000;

%fs=3000;

T=1/fs;

%可选采样长度，等价于采样点数

%n=500;

n=1500;

%n=2000;

t=(0:n-1)\*T;

%构建含有至少三个分量的信号

s=0.7\*sin(2\*pi\*50\*t)+sin(2\*pi\*130\*t)+sin(2\*pi\*90\*t);

%添加随机噪声

x=s+2\*randn(size(t));

%绘制创造信号时域图

figure(1);

plot(t(1:n/4),s(1:n/4));

xlabel("Time(s)");

ylabel("Amolitude");

title("Time Domain Signal");

%绘制含噪声信号时域图

figure(4);

plot(t(1:n/4),x(1:n/4));

xlabel("Time(s)");

ylabel("Amolitude");

title("Signal Corrupted with Zero\_Mean Random Noise");

%对时域图进行快速傅里叶变换，输出频谱图

y=fft(s);

%观察含有噪声信号的快速傅里叶变换结果

%y=fft(x);

%计算谱密度

p2=abs(y/n);

%计算单侧频谱

p1=p2(1:n/2+1);

%将双侧频谱变换为单侧后，需要将除两端外幅值加倍

p1(2:end-1)=2\*p1(2:end-1);

f=fs\*(0:n/2)/n;

figure(2);

plot(f,p1);

xlabel("frequency (Hz)");

ylabel("|p1(f)|");

title("single-sided amplitude spectrum of X(t)");

%d对频谱进行一些滤波操作后，使用快速傅里叶逆变换，观察信号变化

%删除小于100Hz的部分

y(1:151)=0;

y(1350:1500)=0;

w=ifft(y);

%取实部输出

w=real(w);

figure(3);

plot(t(1:250),w(1:250));

xlabel("Time(s)");

ylabel("w(t)");

title("Inverse Fast Fourier transform After Signal Processing");